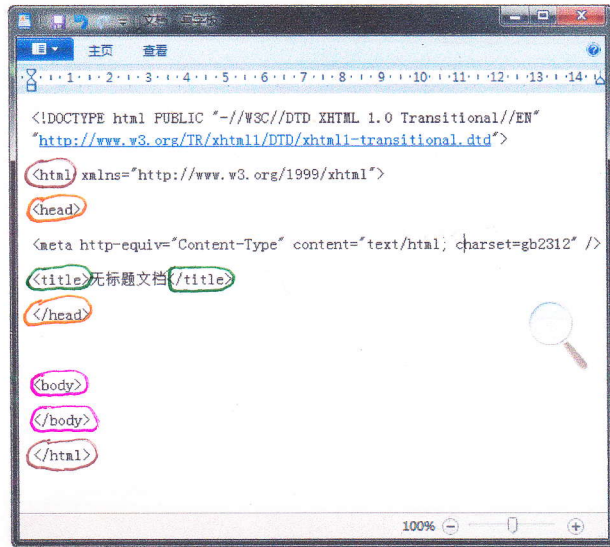
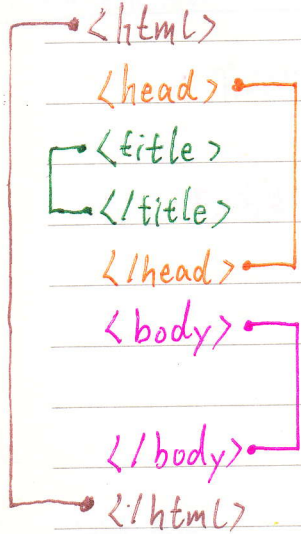


# HTML (Hypertext Markup Language)

Date. sep. 7. 2013 No.

/hazpotelece/

## HTML 结构



\* 每个“开启码”是由“< >”两个符号所框住的。

“关闭码”是由“</ >”所框住的。

每一个HTML码有开就有关。

## 网页颜色设定

Code link.html

```
<body bgcolor = "#xxxxxx" text = "#xxxxxx" link = "#-" vlink = "#-" alink = "#-" >
```

link = 链接 color.

vlink = 已阅读过的 link color

alink = 正在点击的 link color

background = 背景图片 = "back.gif".

• #xxxxxx 这六个数字由 ~~红绿蓝~~ 红 绿 蓝 的 color 元素组成。

为 0~9 及 A-F 组合起来的六个数值。

• 另外，在 xxxxxx 部分也可用颜色的英文名称代替，但 xxxxxx 之前要去掉“#”。

## 加入水平分线

水平分线.html

```
<hr size="5" align="center" noshade width="90%" color="0000ff">
```

size: 线的宽度 align: 控制位置 (left, center, right) width: 控制线长度.

noshade: 控制线无阴影. color: 控制线的颜色.

如果要想在网页中插入水平分线, 直接加入 <hr> 的码即可.

## 清单设定方式

清单.html

### 图头清单

```
<ul>
<li> first
<li> second
<li> third
</ul>
```

在 <ul> 中可加入东西改变效果:

```
<ul type="disc/circle/square">
```

分别是. 实心圆, 空心圆, 方块.

### 数字清单

```
<ol>
<li> things use
<li> things list
<li> things list
</ol>
```

```
<ol type="A/a/I/i">
```

```
<ol start="起始数字">
```

A/a/I/i 为大小写字母及罗马字母.

### 定义清单

```
<dl>
<dt> mouse
<dd> they r animals
<dt> human
<dd> they r smart animals
</dl>
```

## 文字设定

text.html

文字加粗 <b> 文字内容 </b>  
斜体 <i> 文字内容 </i>  
底纹 <u> 文字内容 </u>

```
<font face="Arial" color="#0033ff" size="7">Text</font>
```

要控制文字如上 <font> 码. 即可控制: 字体, 文字色, 大小 "1~7".

<br> 强制换行.

<pre>... </pre> 可将文字格式转化

<p>? 强制换两行. 且可加效果: <p align="left/right/center">Text </p> 59

## 插入超链接

超链接.html.

`<a href = "www.baidu.com" >百度 </a>` 网址超链接  
href: //

`<a href = "index.html" >点我 </a>` 站内文档超链接

可在 `<a href = "url" >` 中插入 `target` 交叉打开关系.

`<a href target = "_new / _top / _blank / Name" >`

相同页面打开 / 不同页面打开

定位 → `<a name = "目标地点" >` 目标地点 `</a>`

插入 href → `<a href = "#目标地点" >` 跳回目标地点 `</a>`

⇒ 文件内连接  
(类似锚点)

## 插入图片

超链接.html.

~~`<img src = "url" >`~~

同时可加入下列属性, 来变化图片.

`<img src = "image/kuala.jpg" >`

属性: width / height / align / border.

## 跑马灯

跑马灯.html.

`<marquee >` 跑马灯 `</marquee >`

加入下列属性可改变效果:

`bgcolor = "#xxxxxx"` 加入背景颜色.

`direction = "left / up / down"` 控制运动方向.

`scrollamount = "数字"` 运动速度

碰到屏幕时

左右碰撞

`behavior = "slide / alternate"` 行进方式.

`width:` } 控制圈行宽度.

`height:` }

# 表格制作

table.html.

```
<table width="300" border="1" cellspacing="2">
```

```
<td>存储1</td>
```

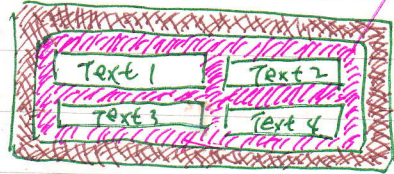
```
<td>存储2</td>
```

```
<tr>
```

```
<td>存储3</td>
```

```
<td>存储4</td>
```

```
</table>
```



width: 控制Table长度可用数字或%

border: 控制外框粗细

cellspacing: 单元格的间隔距离.

background: 背景图片.

align: 水平位置 = left/right/center.

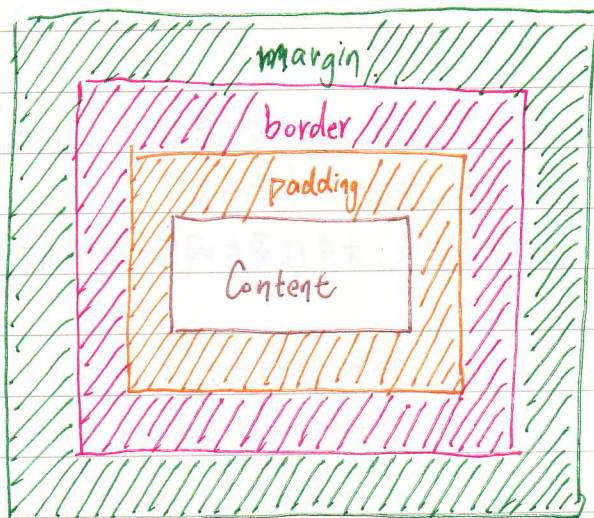
rowspan: 使一单元格下跨 N 个行.

valign: 垂直位置 = top/middle/bottom.

colspan: 使一单元格横跨 N 个栏位.

# 特殊字符

&amp	&
&quot	"
&lt	<
&gt	>
&nbsp	空白



Content: 内容    ~~border: 边框~~    ~~padding: 填充~~    Margin: 间隔.  
padding 填充    border 边框.

## DW 事件

OnLoad: 选定的客体显示在 browser 时.

OnMouse Down 单击 鼠标左键时.

OnUnload: 浏览器退出网页文档时.

on Mouse Move 光标经过选定的要素时.

On Error: 加载网页文档时发生错误时.

on Mouse Out 光标离开选定要素时.

On Blur: 光标移到窗口或框架外时.

on Mouse Over 光标在选定要素上面时.

On Focus: 光标到窗口或框架中激活状态时.

on Mouse up. 放开按住的鼠标左键时.

On click: 用鼠标单击要素时.

on key Down 键盘上某个按键被按下时.

On key up. 放开按下指定的键时.

on key Press 键盘... 被按下并且释放时.

# Navigation bar 案例

navigation bar.html

① `<head>` 和 `</head>` 之间的内容是 head 信息, 一般不显示出来, 但记录了 HTML 文件很多有用的 info, 但之间的 info 你看不到。  
`<body>` 和 `</body>` 之间的信息是正文, 才是你要在网页中展现出来的。

② `<style type="text/css">` `</style>` 表示这是样式表, 在中间放入 CSS 内容。  
`type` 是类型, `text/css` 是要告诉 browser 这里的文本内容 (text) 要当 CSS 来解析。

~~③~~

## 使用 CSS

① 直接添加在 HTML 中  $\Rightarrow$  内嵌样式

`<p style="Color: red; font-size: 25pt">` 实例 `</p>`

② 将 CSS Code 添加在 HTML 的 `<style>` `</style>` 中  $\Rightarrow$  内部样式表

`<head>`

`<style type="text/css">`

`<div {color: sienna;}`

`<div {margin-left: 20px;}`

`body {background-image: url("images/back0.gif");}`

`</style>`

`</head>`

`<body>`

`<div> xxxxxx </div>`

`</body>`

## ③ 导入外部样式表

是指在内部样式表的 `<style>` 里导入一个外部样式表, 用 `@import`.

```
<head>
```

```
<style type="text/css">
```

```
<!--
```

```
@import "mystyle.css";
```

```
-->
```

```
</style>
```

```
</head>
```

## ④ 链入外部样式表

```
<head>
```

```
<link href="mystyle.css" rel="stylesheet" type="text/css" media="all">
```

```
</head>
```

`rel="stylesheet"` 表示在页面中使用这个外部样式表。

`media="all"` 是选则媒体类型, 如: 屏幕, 纸张, 语音合成设备, 盲文阅读设备。

## CSS 选则器

### I: 基本选则器:

#### ① 通配符选择器 (\*)

```
*{
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
demo * { border: 1px solid blue; }
```

所有元素的 margin 和 padding 都为 0.

只影响 div, demo 下的元素, 因此都加上新样式。

```
<div class="demo">
```

```
<ul class="clearfix">
```

```
<li id="first" class="first">1</li>
```

```
<li class="active important">2</li>
```

```
<li class="important items">3</li>
```

```
<li class="important">4</li>
```

```
<li class="items">5</li>
```

```
<li>6</li> <li>7</li> <li>8</li> <li>9</li>
```

```
<li id="last" class="last">10</li>
```

```
</ul>
```

```
</div>
```

## ② 元素选择器: (E)

```
li { background-color: grey; color: orange; }
```

是最常见且基本的，元素有 html, body, p, div 等，

这个 demo 中元素有 div, ul, li 等。

## ③ 类选择器 (. class Name)

- 是以一独立于文本与元素的方式来指定样式。

- 使用类选择器之前需要在 html 元素上定义类名。如：

```
<li class="active important items">2</li>
```

"active important items" 就是给类加上的一个类名

```
.important { font-weight: bold; color: yellow; }
```

```
.items { color: red; }
```

```
.important { color: green; }
```

```
.items { color: #fff; background: #000; }
```

```
.important, items { background: #ccc; }
```

```
.first, last { color: blue; }
```

只对选择器元素同时包含了 "important" 和 "items" 两个类才起作用。

→ 如果一个多类选择器包含的类名中有一个不存在，则无效。列表中没有一个即是。因此又有 last 的写法。



## ④ id 选择器 (#ID)

- 在 HTML 中需加注 ID 名称。
- 在名称前加“#”。

```
# first { background: lime; color: #000 }
```

```
# last { background: #000; color: lime }
```

注意: • 一个文档中一个 id 选择器只能使用一次。

- ID 选择器不能像类选择器一样多本合并使用, 一个元素只能命名一个 ID 名。
- 可在不同的文本中使用相同的 ID 名。

## ⑤ 后代选择器 (E F)

- 前面 E 为祖先元素, F 为后代元素, 选到了 E 元素的所有后代 F 元素。
- 中间需要一个空格

```
demo li { color: blue }
```

① 子元素选择器 (E > F) ~~ul > li~~ ul > li { background: green; color: yellow; }

② 相邻兄弟元素选择器 (E + F)

- 可以选则紧接在另一元素后面的元素, 而且具有一个相同父元素。

```
li + li { background-color: green; color: yellow; border: 1px solid #ccc; }
```

```
.active + li { background: green; ... }
```

③ 通用兄弟选择器 (E ~ F)

- 将选则某元素后所有的所有兄弟元素。
- 需要在同一父级元素中。

```
.active ~ li { background: green; color: yellow; border: 1px }
```

④ 群组选择器: (selector 1, selector 2, ..., selector N)

- 将具有相同样式的元素放在一起, 每个选择器用“,”隔开。

```
.first, .last { background: green; color: yellow }
```